

HOSTEL MANAGEMENT SYSTEM

Submitted in partial fulfilment of the required for the award of

Bachelor of Computer Science

By

A. POORNA CHANDRU

(REGISTER NO 40290067)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE

AND TECHNOLOGY

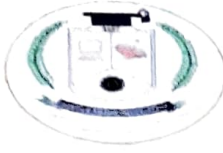
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI

600 119

MAY 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A. POORNA CHANDRU (40290067)** who carried out the project entitled "**HOSTEL MANAGEMENT SYSTEM**" under our supervision from JANUARY 2023 to MAY 2023.

[Signature] 09/05/2023

INTERNAL GUIDE

Dr. V. ULAGAMUTHALVI M.E., Ph.D.,

[Signature]

DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on

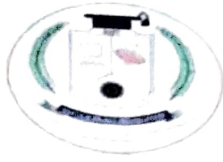
09/05/2023

[Signature] 09/05/2023

Internal Examiner

[Signature] 09/05/2023

External Examiner



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A. POORNA CHANDRU (40290067)** who carried out the project entitled "**HOSTEL MANAGEMENT SYSTEM**" under our supervision from JANUARY 2023 to MAY 2023.

6/8/2023

INTERNAL GUIDE

Dr. V. ULAGAMUTHALVI M.E., Ph.D.,

2

Rekha

DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on

6/9/2023

Internal Examiner

09/05/2023

13/05/2023

External Examiner

DECLARATION

I am, **A. Poorna Chandru (40290067)** here by declare that the Project Report entitled **"HOSTEL MANAGEMENT SYSTEM"** done by us under the guidance of **Dr. V. Ulagamuthalvi M.E., Ph.D.,** Professor, Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY,** Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of bachelor of Science degree in Computer Science.

DATE: 09/05/2023

PLACE: CHENNAI



A. Poorna Chandru

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D. Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep since of gratitude to my project guide **Dr. V. Ulagamuthalvi M.E., Ph.D., Professor** of Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

A. Poorna Chandru

ABSTRACT

For the past few years the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more users friendly and more GUI oriented. The proposed web application helps in the development of the hostel booking system with verification of rooms available through online processing. The information will be updated by the admin where the room details, fees details, food details, further complaints will be taken care by the admin. The student makes an access with a registration and login system which further leads to identifying the hostel rooms available, attendance marked food etc. The hostel web application gives out a process of making their complaints to the admin if it is so.

The proposed system gives out exact solution to the problems and maintains a server for easy retrieval. The developed system will manage the allotment of rooms for the students and the secure payment of their fees. The developed web application shows a good user interface and the processing speed compared with manual works. Secure payment shows a good accessing system and a reliable trustworthy service is enhanced.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	VII
	LIST OF FIGURES	VI
1	INTRODUCTION	8
2	LITERATURE SURVEY	9
	2.1 ONLINE HOSTEL MANAGEMENT SYSTEM	9
	2.2 HOSTEL MANAGEMENT SYSTEM	9
3	SYSTEM ANALYSIS	11
	3.1 FEASIBILITY STUDY	11
	3.2 TECHNICAL FEASIBILITY	11
	3.3 OPERATIONAL FEASIBILITY	11
	3.4 ECONOMIC FEASIBILITY	12
	3.5 EXISTING SYSTEM	12
	3.6 PROPOSED SYSTEM	12
4	SYSTEM SPECIFICATION	14
5	SYSTEM DESCRIPTION	15
6	SYSTEM DESIGN	20
	6.1 ARCHITECTURE DIAGRAM	20
	6.2 DATA FLOW DIAGRAM	21

ABSTRACT

For the past few years the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more users friendly and more GUI oriented. The proposed web application helps in the development of the hostel booking system with verification of rooms available through online processing. The information will be updated by the admin where the room details, fees details, food details, further complaints will be taken care by the admin. The student makes an access with a registration and login system which further leads to identifying the hostel rooms available, attendance marked food etc. The hostel web application gives out a process of making their complaints to the admin if it is so.

The proposed system gives out exact solution to the problems and maintains a server for easy retrieval. The developed system will manage the allotment of rooms for the students and the secure payment of their fees. The developed web application shows a good user interface and the processing speed compared with manual works. Secure payment shows a good accessing system and a reliable trustworthy service is enhanced.

TABLE OF CONTENT

PTER NO	TITLE	PAGE NO
	ABSTRACT	VII
	LIST OF FIGURES	VI
1	INTRODUCTION	8
2	LITERATURE SURVEY	9
	2.1 ONLINE HOSTEL MANAGEMENT SYSTEM	9
	2.2 HOSTEL MANAGEMENT SYSTEM	9
3	SYSTEM ANALYSIS	11
	3.1 FEASIBILITY STUDY	11
	3.2 TECHNICAL FEASIBILITY	11
	3.3 OPERATIONAL FEASIBILITY	11
	3.4 ECONOMIC FEASIBILITY	12
	3.5 EXISTING SYSTEM	12
	3.6 PROPOSED SYSTEM	12
4	SYSTEM SPECIFICATION	14
5	SYSTEM DESCRIPTION	15
6	SYSTEM DESIGN	20
	6.1 ARCHITECTURE DIAGRAM	20
	6.2 DATA FLOW DIAGRAM	21

	6.3 CLASS DIAGRAM	26
	6.4 USE CASE DIAGRAM	27
	6.5 SEQUENCE DIAGRAM	28
	6.6 ACTIVITY DIAGRAM	30
7	IMPLEMENTATION	32
	7.1 MODULE DESCRIPTION	32
8	SYSTEM TESTING	33
	8.1 TEST RESULT	33
	8.2 PSEUDOCODE	37
	8.3 RESULT/FINDING	37
9	CONCLUSION	38
	9.1 CONCLUSION	38
	9.2 FUTURE SCOPE	38
10	REFERENCE	39
11	APPENDIX	40
	11.1 SOURCE CODE	40
	11.2 OUTPUT	68

LIST OF FIGURES

FIGURE NO	FIGURE NAME
6.1	SYSTEM ARCHITECTURE
6.2	DATA FLOW DIAGRAM
6.3	FLOW DIAGRAM LEVEL 0
6.4	FLOW DIAGRAM LEVEL 1
6.5	CLASS DIAGRAM
6.6	USE CASE DIAGRAM
8.1	UNIT TESTING FOR THE ADMIN LOGIN
8.2	SYSTEM TESTING OF ADDS DETAILS
8.3	INTEGRATION TESTING WITH VIEW BOOKING PAGE
8.4	THE VALIDATING TESTING FOR THE ADMIN PAGE

CHAPTER 1

INTRODUCTION

The online hostel management system is web based software to provide college students accommodation to the university hostel more efficiently. This project also keeps details of the hostellers and applied students. It is headed by Warden. He will be the administrator. For accommodate a large number of students into hostel. In our current era of automated systems with it being either software or hardware, it's not advisable to be using manual system. Hostels without a management system are usually done manually. Registration forms verification to other data saving processes are done manually and most at times, they are written on paper. Thus a lot of repetitions can be avoided with an automated system. The drawbacks of existing systems lead to the design of a computerized system that will help reduce a lot of manual inputs. With this system in place, we can improve the efficiency of the system, thus overcome the drawbacks of the existing manual system. This system is designed in favor of the hostel management which helps them to save the records of the students about their rooms and other things. It helps them from the manual work from which it is very difficult to find the record of the students and the mess bills of the students, and the information of about the those ones who had left the hostel years before. This system gives an idea about how a student and fee details, room allocation, mess expenditure are maintained in a better way. The hostel management system will also contain special features like how many students are in a room, student's id and free rooms or space available. The administration has a unique identity for each member as well as students details. The hostel management system is web based software to provide college students accommodation to the university hostel more efficiently. This project also keeps details of the hostellers and applied students.

CHAPTER 2

LITERATURE SURVEY

2.1. ONLINE HOSTEL MANAGEMENT SYSTEM

"ONLINE HOSTEL MANAGEMENT SYSTEM" is software developed for managing various activities in the hostel. For the past few years the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

2.2. HOSTEL MANAGEMENT SYSTEM

As the name specifies "HOSTEL MANAGEMENT SYSTEM" is a software developed

for managing various activities in the hostel. For the past few years the number of educational institutions are increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system Which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

- Less human error

- Strength and strain of manual labour can be reduced
- High security
- Data redundancy can be avoided to some extent
- Data consistency
- Easy to handle
- Easy data updating
- Easy record keeping
- Backup data can be easily generated

HOSTEL MANAGEMENT SYSTEM FOR STUDENT MAINTENANCE

"HOSTEL MANAGEMENT SYSTEM" is software created to manage numerous operations in the hostel, as the name implies. The number of educational institutions has been steadily rising during the last few years. As a result, there are more dormitories available to house the students that attend this college. As a result, the person in charge of the hostel is under a lot of stress, and software is not typically used in this situation. This specific project addresses hostel management issues while avoiding issues that arise when tasks are performed manually. The construction of a computerised hostel administration system that is compatible with the existing system and is more user-friendly and GUI focused results from the identification of the shortcomings of the existing system. The shortcomings of the current hostel management system can be fixed by increasing system efficiency. less human mistake, lower strength and strain from manual labour, extreme security, Redundant data can be prevented to some extent, Consistency of the data, ease of handling, ease of data upgrading, and ease of record keeping Data backups are simple to create.

3.4 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost/ benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

3.5 EXISTING SYSTEM

The existing system is manual based and need lot of efforts and consume enough time. In the existing system we can apply for the hostels online but the allotment processes are done manually. It may lead to corruptions in the allocation process as well as hostel fee calculation. The existing system does not deals with mess calculation and complaint registration.

DISADVANTAGES

- More human power
- More strength and strain of manual labour needed
- Repetition of same procedure.
- Low security.
- Data redundancy.
- Difficulty to handle.
- Difficulty to update data. Record keeping is difficult.
- Backup data can be easily generated.

3.6 PROPOSED SYSTEM

The proposed system is having many advantages over the existing system. It require less overhead and very efficient. The proposed system deals with the mess calculation and room allotment process efficiently.

ADVANTAGES

- Save time on admin tasks.
- Develop strong relationships with your guests.
- Increase your online visibility.
- Implement an effective revenue management system.

- Manage distribution functions.
- Increase bookings.
- Accurate daily report

CHAPTER 4

SYSTEM SPECIFICATION

HARDWARE REQUIREMENTS

- Processor - AMD PRO
- Hard disk capacity - 500 GB
- Internal memory capacity - 4 GB
- Keyboard - STANDARD KEYBOARD
- Cpu clock - 1.08 GHz
- System Type - 64 bit Operating System

SOFTWARE REQUIREMENTS

- Operating system - Windows10
- Front end - PHP
- Database - MYSQ

CHAPTER 5

SOFTWARE DESCRIPTION

FRONT END (PHP):

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994. PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire ecommerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time. PHP is forgiving: PHP language tries to be as forgiving as possible. PHP Syntax is C-Like.

Common Uses of PHP

PHP performs system functions, i.e. from files on a system it can create, open, read, write and close them. The other uses of PHP are:

PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user. You add, delete and modify elements within your database thru PHP. Access cookies variables and set cookies. Using PHP, you can restrict users to access some pages of your website. It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

PHP Variables

The main way to store information in the middle of a PHP program is by using a variable. Here are the most important things to know about variables in PHP.

- All variables in PHP are denoted with a leading dollar sign (\$).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types - a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.

PHP variables are Perl-like. PHP has a total of eight data types which we use to instruct our variables:

- **Integers:** are whole numbers, without a decimal point, like 4195.
- **Doubles:** are floating-point numbers, like 3.14159 or 49.1.
- **Booleans:** have only two possible values either true or false.
- **NULL:** is a special type that only has one value: NULL.
- **Strings:** are sequences of characters, like 'PHP supports string operations.'
- **Arrays:** are named and indexed collections of other values.
- **Objects:** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources:** are special variables that hold references to resources external to PHP (such as database connections).

ack End (MySQL):

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central

component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), ImagebookTwitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, which enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

Command line

MySQL ships with some command line tools. Third-parties have also developed tools to manage a MySQL server, some listed below. Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up

servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Programming. MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO Open Server, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.

MySQL is written in C and C++. Its SQL parser is written in yacc, and a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interimage called MyODBC allows additional programming languages that support the ODBC inter image to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

Features

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licenses. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- Information schema
- Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
- X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine
- Transactions with the InnoDB, and Cluster storage engines

- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- Full-text indexing and searching using MyISAM engine
- Embedded database library
- Partititoned tables with pruning of partitions in optimiser
- Shared-nothing clustering through MySQL Cluster
- Hot backup (via mysqlhotcopy) under certain conditions
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time): Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, EXAMPLE, Maria, and InnoDB, which was made the default as of 5.5). Partner-developed storage engines (solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.

CHAPTER 6

SYSTEM DESIGN

6.1 ARCHITECTURE DIAGRAM

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software.

An architecture diagram is a graphical representation of a set of concepts that are part of architecture, including their principles, elements and components. Architecture diagram can help system designers and developers visualize the high-level, overall structure of their system or application, in order to ensure the system meets their users' needs. Using architecture diagram, you can also describe patterns that are used throughout the design. It's somewhat like a blueprint that you use as a guide, so that you and your colleagues can discuss, improve and follow.

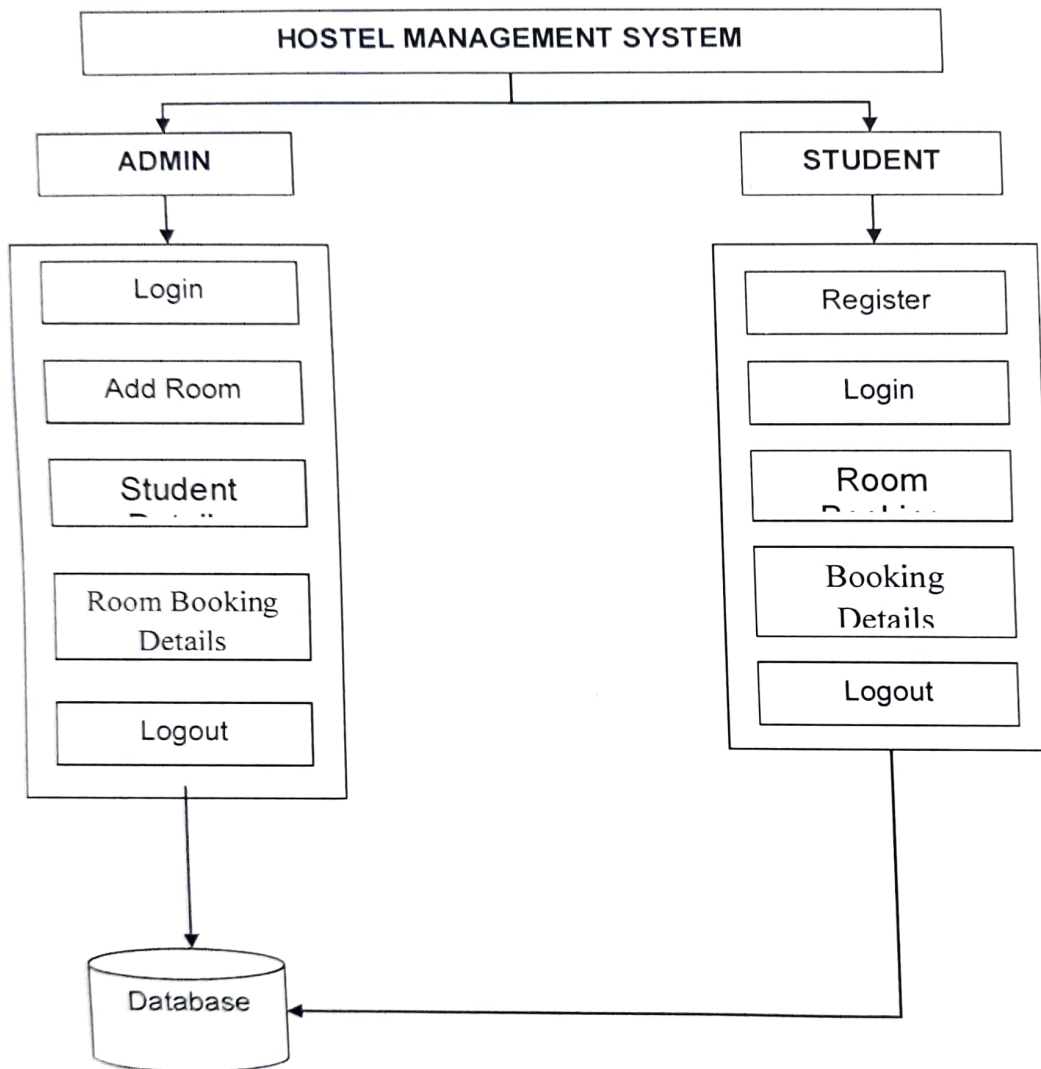


Fig 6.1 System Architecture

6.2 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must

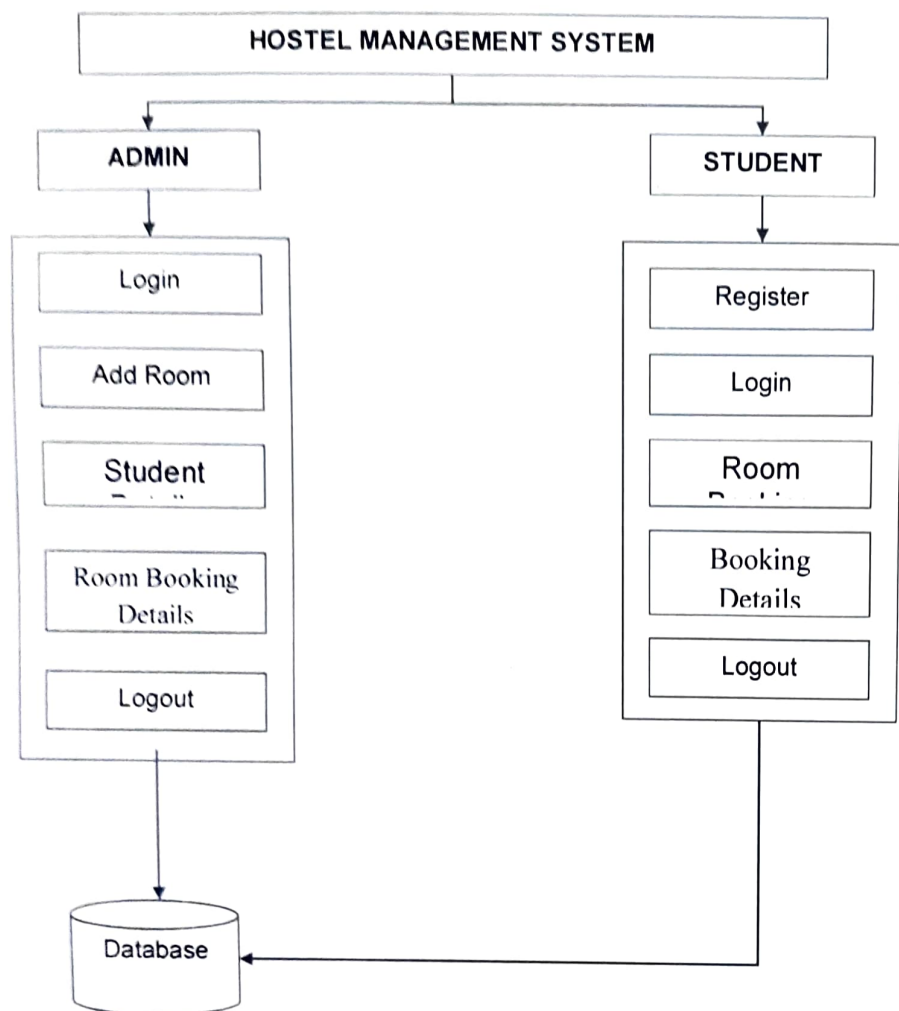


Fig 6.1 System Architecture

6.2 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must

identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols:

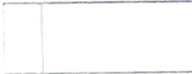
Symbol	Description
	An entity . A source of data or a destination for data.
	A process or task that is performed by the system.
	A data store , a place where data is held between processes.
	A data flow .

Fig 6.2 Data Flow Symbols

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

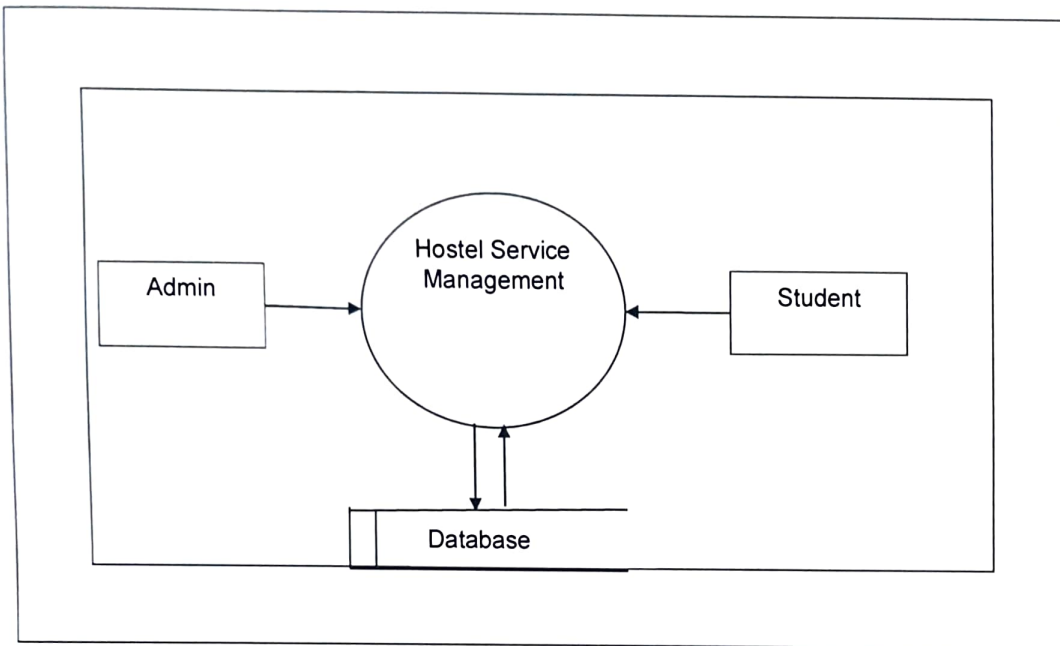


Fig 6.3 Flow Diagram Level 0

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into sub processes that can then be analyzed and improved on a more intimate level.

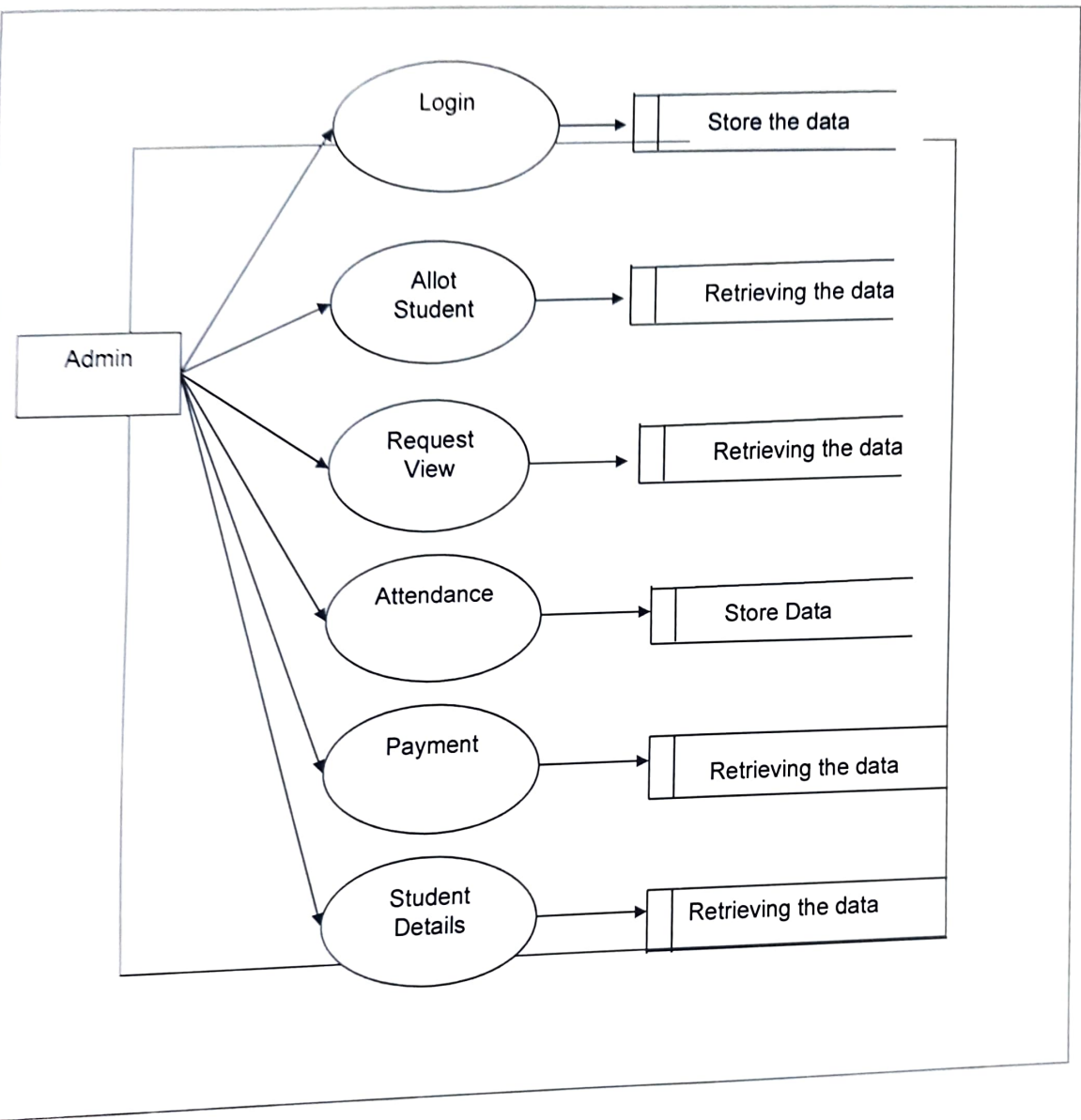
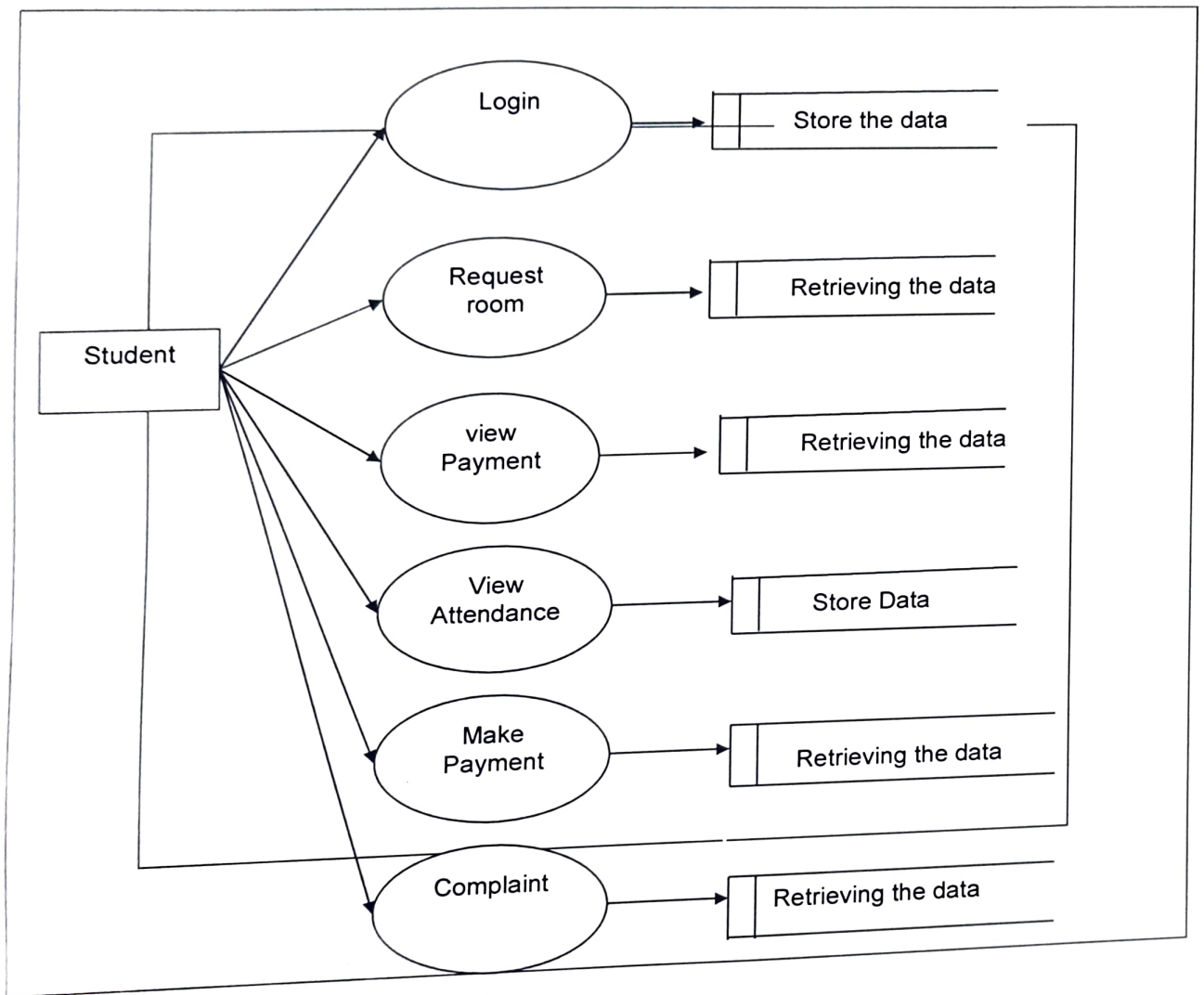
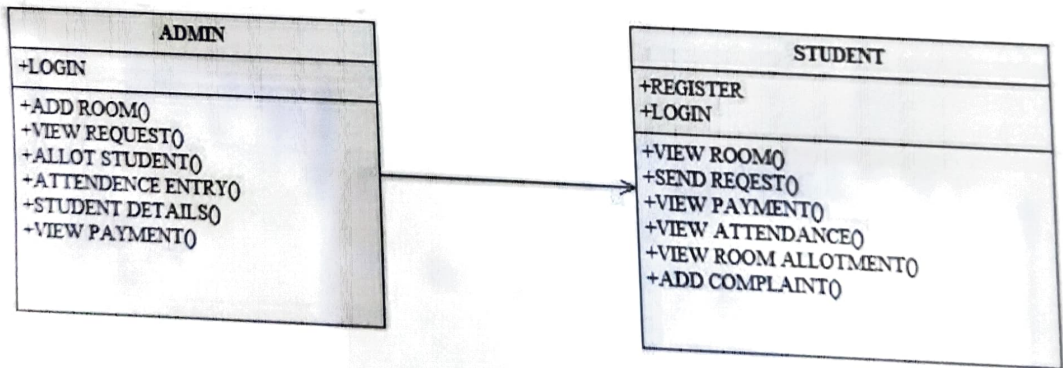


Fig 6.4 Flow Diagram Level 1



6.3 CLASS DIAGRAM

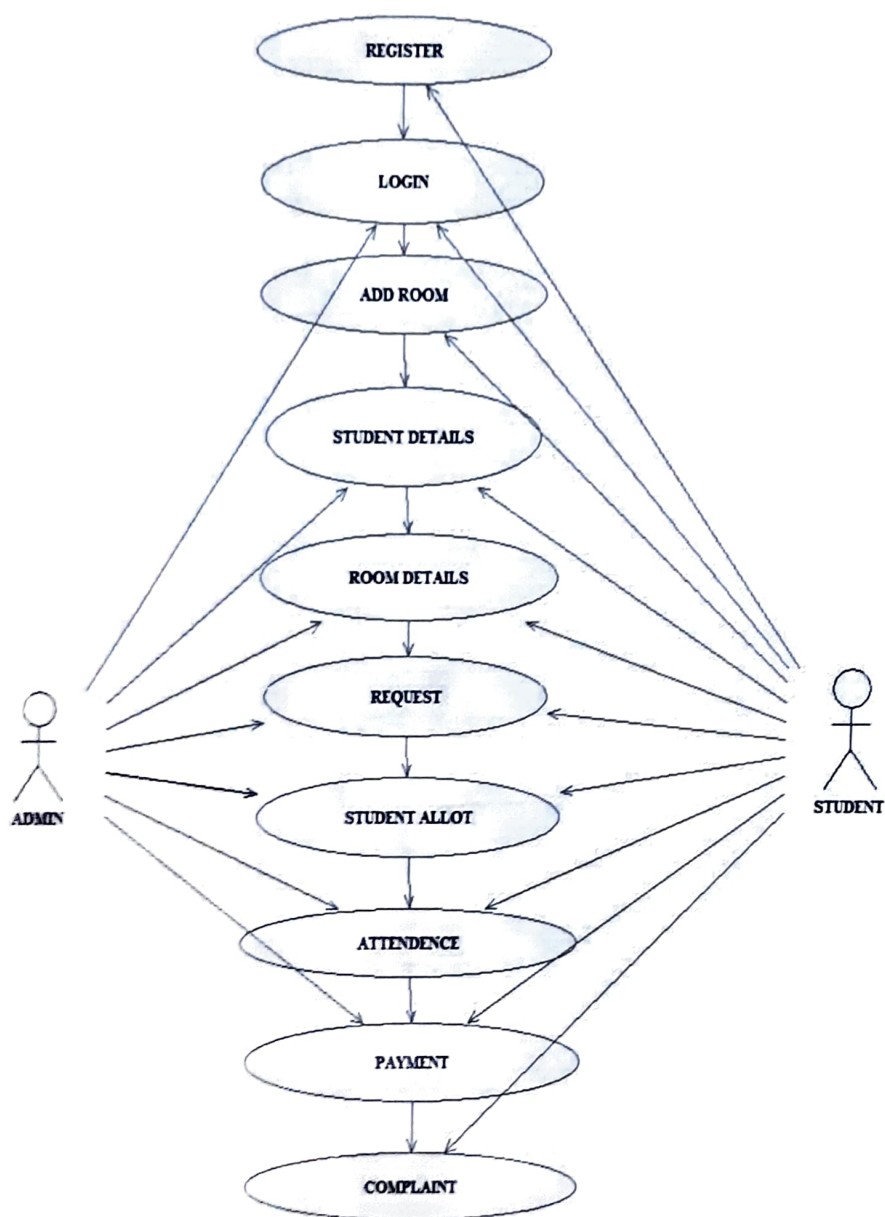
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed. In the diagram, classes are represented with boxes that contain three compartments: The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.



6.4 USE CASE DIAGRAM

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor can be a human or other external system.

An actor in the Unified Modeling Language (UML) "specifies a role played by a user or any other system that interacts with the subject." "An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject." UML Use Case Include. Use case include is a directed relationship between two use cases which is used to show that behavior of the included use case (the addition) is inserted into the behavior of the including (the base) use case.

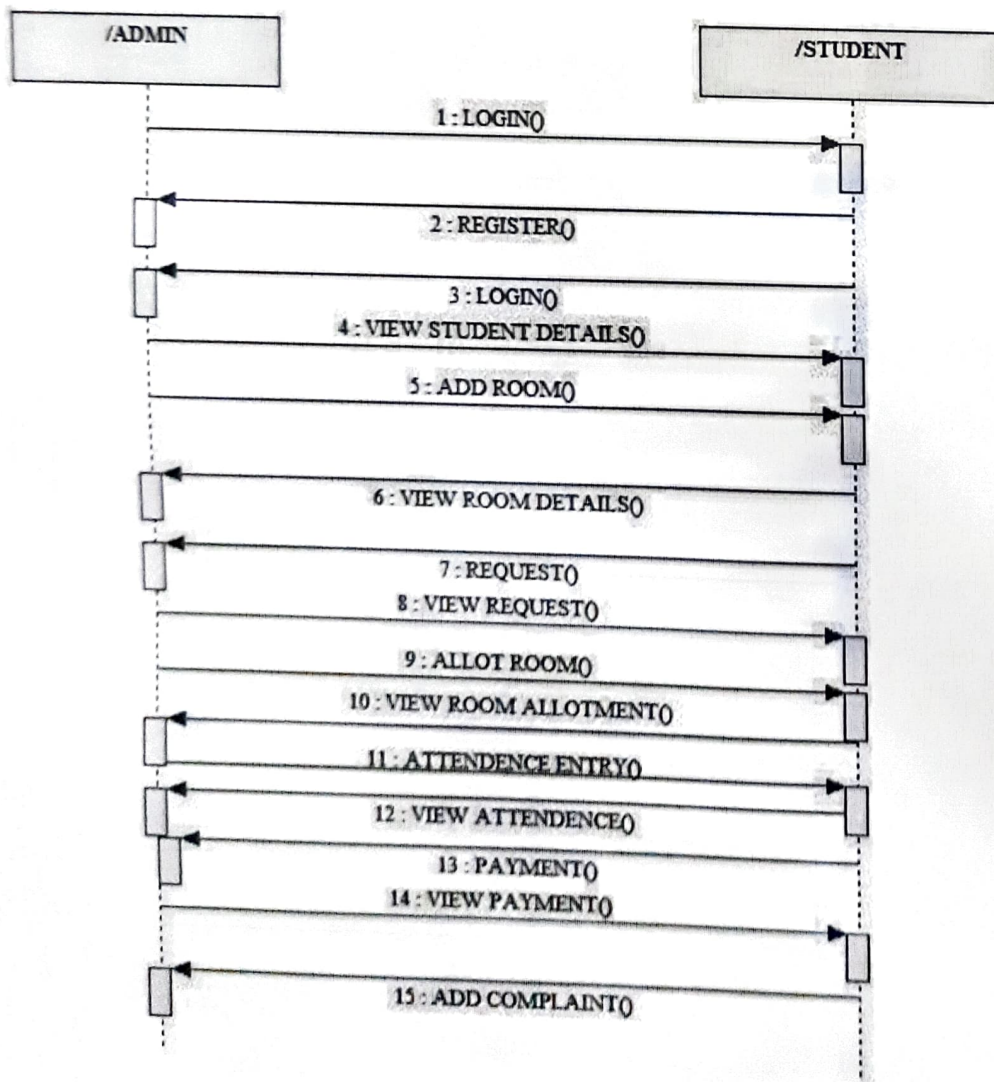


6.5 SEQUENCE DIAGRAM

Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time

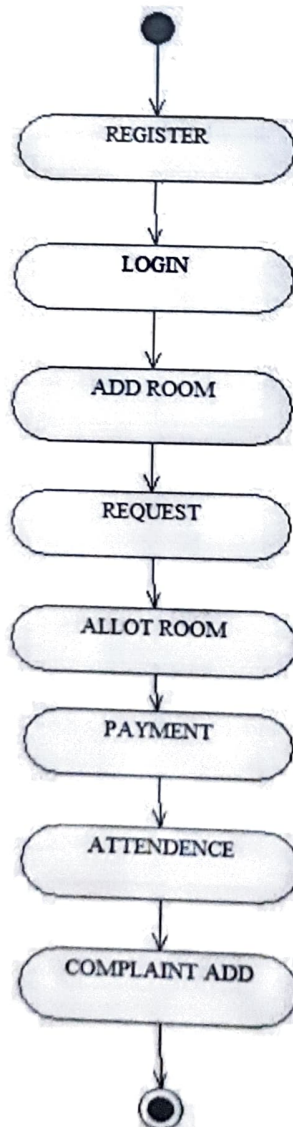
sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. Objects as well as classes can be targets on a sequence diagram, which means that messages can be sent to them.



6.6 ACTIVITY DIAGRAM

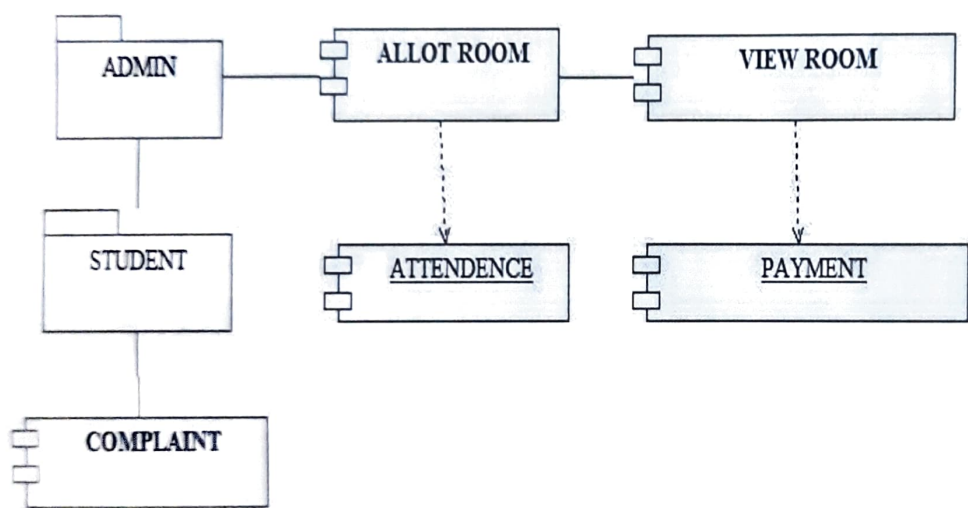
Activity diagram is to describe the dynamic aspects of the system. Activity diagram is

basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling.



COMPONENT DIAGRAM

A component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



CHAPTER 7

IMPLEMENTATION

7.1 MODULE DESCRIPTION

ADMIN

- Admin Login – The admin will make a login with the default username and password which are provided in the server side
- Admin Add Room – The admin will add the details of the hostel room's availability where the admin will get the details of the processing information's with image.
- Admin Request- The admin will get the details of the room based request here the room will be allotted as per order by the students the remaining students need out to wait for the room or they need to take another room
- View Payment – The admin in this module will view the payment details made by the student for the hostel fees will be viewed.
- View attendance – The admin will view the attendance of the students with their check in and check out

STUDENT

- Student Register – The student in this module will make a registration system. The registration works with the high processing element
- Student Login – The student will give a login with the generated username and password system
- Student Book Room- The student in this module will book the room and wait for the confirmation from the admin. The student need to wait if the room is already booked by other one
- Student Payment Alert – The student will make an online payment on their hostel fees and get a bill through online itself. They will get the payment details and the alert for the payment will be done
- Student Complaint – In this module, the student are provided with a session where they can make complain

CHAPTER 8

SYSTEM TESTING

8.1 TEST RESULTS

Testing

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

Following are the typical design techniques in software engineering:

Deriving test cases directly from a requirement specification or black box test design technique. The Techniques include:

- Boundary Value Analysis (BVA)
- Equivalence Partitioning (EP)
- Decision Table Testing
- State Transition Diagrams
- Use Case Testing

2. Deriving test cases directly from the structure of a component or system:

- Statement Coverage
- Branch Coverage
- Path Coverage
- LCSAJ Testing

3. Deriving test cases based on tester's experience on similar systems or testers intuition:

- Error Guessing
- Exploratory Testing

Test Reports

1. Unit testing

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.



Admin Login

Username	admin
Password	*****
Login	

[Unit testing for the admin Login]

2. System testing

Several modules constitute a project. If the project is long-term project, several developers write the modules. Once all the modules are integrated, several errors may arise. The testing done at this stage is called system test. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. Testing a specific

hardware/software installation. This is typically performed on a COTS (commercial off the shelf) system or any other system comprised of disparate parts where custom configurations and/or unique installations are the normal.

Hostel Service Management

[Home](#) [Add Room](#) [View Request](#) [Allot Student](#) [Attendance Entry](#) [Student Details](#) [View Payment](#) [Logout](#)

Add Room

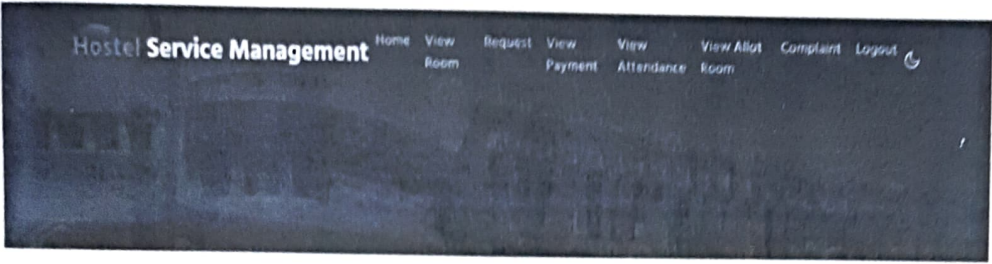
Room No	5001
Total Members	4
Space Available	500x500
Total Beds	4
Fans	4
Lights	4
Fees	5000
Food Availability	☒ YES
Image	Choose File a2.jpg

Add Room

[System testing of adds Room details]

3. Integration testing

Testing is which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.



Student Home

ID	Room No	Total Members	Space Available	Total Beds	Fans	Lights	Fees Details	Food Availability	Image	Request
1	5001	4	500x600	4	4	4	5000	YES		Click...

[Integration testing with view booking page]

4. Validation Testing

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs.



Admin Login

Username

Password

Please fill out this field

Login

[The validation testing for the admin login]

8.2 PSEUDOCODE

REGISTRATION DETAILS

- Step 1: Start
- Step 2: Enter the Name
- Step 3: Enter the Address
- Step 4: Enter the Contact
- Step 5: Enter the E-mail
- Step 6: Enter the user name
- Step 7: Enter the password
- Step 8: Stop

STUDENT REQUEST DETAILS

- Step 1: Start
- Step 2: Enter the student name
- Step 3: Enter the send request
- Step 4: Enter the room details
- Step 5: Enter the allotment details
- Step 6: Stop

8.3 RESULTS /FINDING

The developed project was tested with the dell laptop with 2 MP camera and which gives the promising output even for the user who is in the distance about 10 feet as well as even though the face of the person appears too small because he is standing away from camera this project is able to detect his face and retrieve his name accurately with minimum delay.

CHAPTER 9

CONCLUSION

9.1 Conclusion

Online Hostel Service Management System is very useful for hostel allotment and fee calculation. This hostel management software is designed for people who want to manage various activities in the hostel. For the past few years the numbers of educational institutions are increasing rapidly. Thereby the numbers of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

9.2 Future Scope

In future system, still more more user-friendly, graphical-user-interface oriented, reliable, efficient and secured with access control mechanisms.

CHAPTER 10

REFERENCES

BOOK REFERENCE:

- [1] Jamie Chan, "Learn PHP in One Day and Learn It Well", 23 June 2020
- [2] Lambert M. Surhone, Miriam T. Timpledon , "Php: Hypertext Preprocessor", 2010
- [3] Paul Whitehead, Joel Desamero , "PHP-Your Visual Blueprint for Creating Open Source, Server-side Content", 2001
- [4] Leon Atkinson, Zeev Suraski , "Core PHP Programming", 2004
- [5] Peter Lavin , "Object-oriented PHP, Concepts, Techniques, and Code", 2006

WEB REFERENCE:

- [1] https://www.w3schools.com/php/php_intro.asp
- [2] <https://en.wikipedia.org/wiki/PHP>
- [3] https://www.tutorialspoint.com/php/php_introduction.html

CHAPTER 11
APPENDIX
SOURCE CODE

Admin

```
<?php
extract($_POST);
session_start();
include("dbconnect.php");
if(isset($_POST['btn']))
{
    $uname=$_REQUEST['username'];
    $password=$_REQUEST['password'];

    if($conn->connect_error)
    {
        die("Connection failed: " . $conn->connect_error);
    }
    else
    {
        $qry="select * from admin where username='$uname' and password='$password'";
        $result = mysqli_query($conn, $qry);

        if (mysqli_num_rows($result))
        {
            ?>
            <script language="javascript">
            alert("login Success");
            window.location.href="admin_home.php";
            </script>
            <?php

        }
    }
}
```

```

else
{
?>
<script language="javascript">
alert("Invalid user");
</script>
<?php

}
}
$conn->close();
}
?>

<!--
Author: W3layouts
Author URL: http://w3layouts.com
-->

<!doctype html>
<html lang="zxx">

<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<meta name="keywords"
content="Atelier Responsive web template, Bootstrap Web Templates, Android
Compatible web template, Smartphone Compatible web template, free webdesigns for
Nokia, Samsung, LG, SonyEricsson, Motorola web design" />
<title>Hostel Management | Admin </title>
<link
href="//fonts.googleapis.com/css2?family=Hind+Siliguri:wght@400;500;600;700&display=
swap"

```

```

    rel="stylesheet">
<!-- Template CSS -->
<link rel="stylesheet" href="assets/css/style-starter.css">
</head>

<body>
    <!--header-->
    <header id="site-header" class="w3lhny-head fixed-top">
    <div class="container">
        <nav class="navbar navbar-expand-lg stroke px-0">
            <h1> <a class="navbar-brand" href="#">
                <span class="sublog">Hostel </span> Service Management
            </a></h1>
            <!-- if logo is image enable this
            <a class="navbar-brand" href="#index.html">
                
            </a> -->
            <button class="navbar-toggler collapsed bg-gradient" type="button" data-
toggle="collapse"
                data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-
expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon fa icon-expand fa-bars"></span>
                <span class="navbar-toggler-icon fa icon-close fa-times"></span>
            </button>

            <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
                <ul class="navbar-nav mx-lg-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="index.php">Home <span class="sr-
only">(current)</span></a>
                    </li>
                    <li class="nav-item ">
                        <a class="nav-link" href="admin.php">Admin</a>
                    </li>
                    <li class="nav-item">

```

```
<a class="nav-link" href="student.php">Student</a> </li>
```

```
</ul>
```

```
<!--/search-right-->
```

```
</div>
```

```
<!-- toggle switch for light and dark theme -->
```

```
<div class="mobile-position">
```

```
<nav class="navigation">
```

```
<div class="theme-switch-wrapper">
```

```
<label class="theme-switch" for="checkbox">
```

```
<input type="checkbox" id="checkbox">
```

```
<div class="mode-container">
```

```
<i class="gg-sun"></i>
```

```
<i class="gg-moon"></i>
```

```
</div>
```

```
</label>
```

```
</div>
```

```
</nav>
```

```
</div>
```

```
<!-- //toggle switch for light and dark theme -->
```

```
</nav>
```

```
</div>
```

```
<!-- search -->
```

```
<!-- //search -->
```

```
</header>
```

```
<!--/header-->
```

```
<!--/inner-page-->
```

```
<div class="inner-banner">
```

```
</div>
```

```
<section class="w3l-breadcrumb">
```

```

</section>
<!--//inner-page-->
<!-- about page about section -->
<section class="w3l-index3" id="about">
  <div class="midd-w3 py-5">

    <div class="w3l-header-sec text-center mb-4">

      <h3 class="title-w3l mx-0">Admin Login</h3>
      <p class="title-w3l mx-0">&nbsp;</p>
      <form name="form1" method="post" action="">
        <table width="329" height="127" border="0" align="center">
          <tr>
            <td>UserName</td>
            <td><input name="username" type="text" id="username"
placeholder="UserName" required=""></td>
          </tr>
          <tr>
            <td>Password</td>
            <td><label>
              <input name="password" type="password" id="password"
placeholder="Password" required="">
            </label></td>
          </tr>
          <tr>
            <td>&nbsp;</td>
            <td><label>
              <input name="btn" type="submit" id="btn" value="Login" class="ar_btn">
            </label></td>
          </tr>
        </table>
      </form>
    </div>

```

```

</section>
<!--//inner-page-->
<!-- about page about section -->
<section class="w3l-index3" id="about">
  <div class="midd-w3 py-5">

    <div class="w3l-header-sec text-center mb-4">

      <h3 class="title-w3l mx-0">Admin Login</h3>
      <p class="title-w3l mx-0">&nbsp;</p>
      <form name="form1" method="post" action="">
        <table width="329" height="127" border="0" align="center">
          <tr>
            <td>UserName</td>
            <td><input name="username" type="text" id="username"
placeholder="UserName" required=""></td>
          </tr>
          <tr>
            <td>Password</td>
            <td><label>
              <input name="password" type="password" id="password"
placeholder="Password" required="">
</label></td>
          </tr>
          <tr>
            <td>&nbsp;</td>
            <td><label>
              <input name="btn" type="submit" id="btn" value="Login" class="ar_btn">
</label></td>
          </tr>
        </table>

```

OUTPUT



Admin Login

UserName	admin
Password	*****
Login	

localhost says
login success

OK

Hostel Service Management

Home Add Room View Request Allot Student Attendance Entry Student Details View Payment Logout

Add Room

Room No

Total Members

Space Available

Total Beds

Fans

Lights

Fees

Food Availability ☒

Image: a2.jpg

Add Room

localhost says
login success

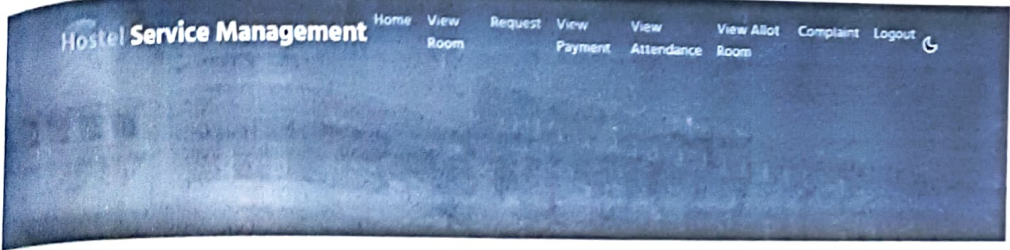
OK

Hostel Service Management

[Home](#)[View
Room](#)[Request](#)[View
Payment](#)[View
Attendance](#)[View Allot
Room](#)[Complaint](#)[Logout](#)

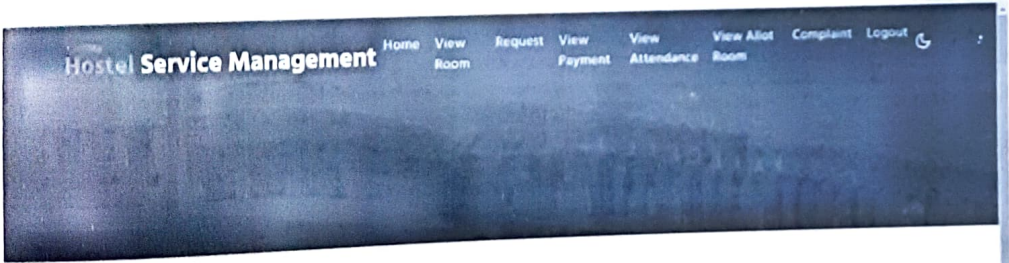
Student Home

ID	Room/No	Total Members	Space Available	Total Beds	Fans	Lights	Fess Details	Food Availability	Image	Request
1	5001	4	500x500	4	4	4	5000	YES		Click...



Allot Room

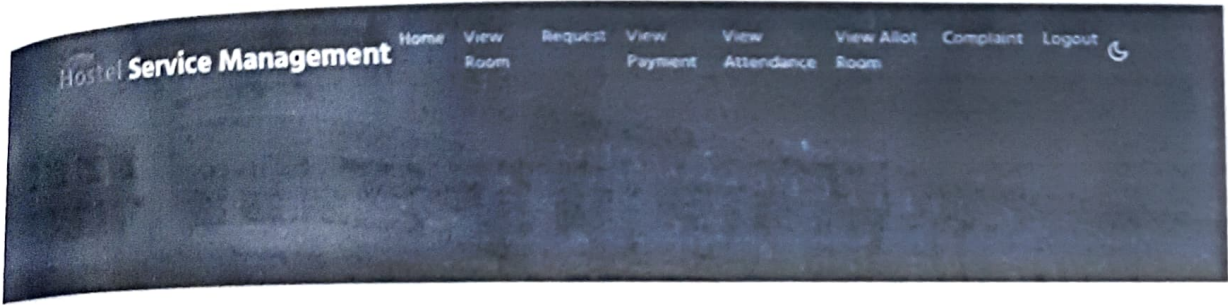
ID	Name	Room No	Address	Contact	Email_Id
1	hendry	5001	trichy	9876543210	hendry@gmail.com



Complaint Box

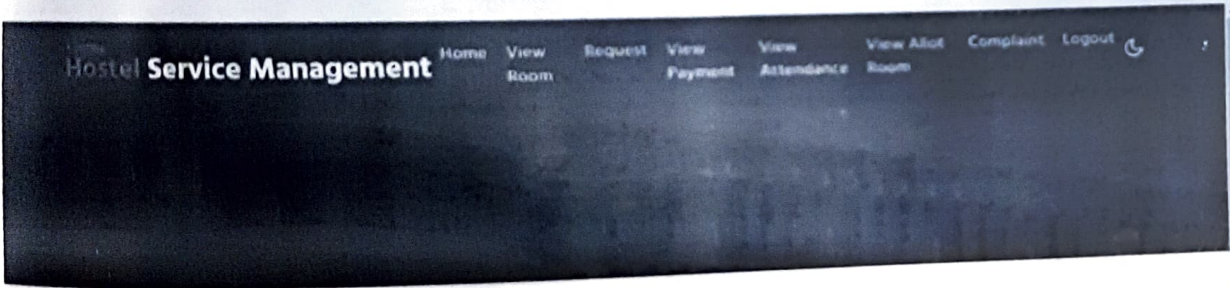
Complaint

Id	User	Complaint	Reply	Status
----	------	-----------	-------	--------



Allot Room

ID	Name	Room No.	Address	Contact	Email_Id
1	hendry	5001	trichy	9876543210	hendry@gmail.com



Complaint Box

Complaint

Id	User	Complaint	Reply	Status
----	------	-----------	-------	--------